

---

## Cubes Convexes

**Sébastien Nedjar — Alain Casali — Rosine Cicchetti — Lotfi Lakhal**

*Laboratoire d'Informatique Fondamentale de Marseille (LIF), CNRS UMR 6166  
Université de la Méditerranée, Case 901  
163 Avenue de Luminy, 13288 Marseille cedex 9  
{nedjar, casali, cicchetti, lakhal}@lif.univ-mrs.fr*

---

*RÉSUMÉ. De nombreuses approches ont proposé de pré-calculer des cubes de données afin de répondre efficacement aux requêtes OLAP. La notion de cube de données a été déclinée de différentes manières : cubes icebergs, cubes intervallaires ou encore cubes différentiels. Dans cet article, nous introduisons le concept de cube convexe qui permet de capturer tous les tuples d'un cube de données satisfaisant une combinaison de contraintes monotones/antimonotones et peut être représenté de façon très compacte de manière à optimiser à la fois le temps de calcul et l'espace de stockage nécessaire. Le cube convexe n'est pas une structure « de plus » à ajouter à la liste des variantes du cube, mais nous le proposons comme une structure unificatrice caractérisant, de manière simple, solide et homogène, les autres types de cubes cités. Enfin, nous proposons une nouvelle variante : le cube émergent qui met en évidence les renversements significatifs de tendances. Nous en proposons une représentation compacte et cohérente avec les caractérisations précédentes.*

*ABSTRACT. In various approaches, data cubes are pre-computed in order to answer efficiently OLAP queries. The notion of data cube has been declined in various ways: iceberg cubes, range cubes or differential cubes. In this paper, we introduce the concept of convex cube which captures all the tuples of a datacube satisfying a constraint combination. It can be represented in a very compact way in order to optimize both computation time and required storage space. The convex cube is not an additional structure appended to the list of cube variants but we propose it as a unifying structure that we use to characterize, in a simple, sound and homogeneous way, the other quoted types of cubes. Finally, we introduce the concept of emerging cube which captures the significant trend inversions. characterizations.*

*MOTS-CLÉS : Analyse multi-dimensionnelle, Cubes de données, Cubes convexes, Cubes émergents, Transversaux cubiques*

*KEYWORDS: Multidimensional analysis, Datacubes, Convex cubes, Emergent Cubes, Cube Transversals*

---

## 1. Introduction et motivations

En pré-calculant tous les agrégats possibles à différents niveaux de granularité, les cubes de données (Gray *et al.*, 1997) permettent une réponse efficace aux requêtes OLAP et sont donc un concept clef pour la gestion des entrepôts. Plus récemment, le calcul de cubes a été utilisé avec succès pour l'analyse multidimensionnelle de flots de données (Han *et al.*, 2005). Dans ce type d'applications, d'énormes volumes de données, à un niveau de granularité très fin, sont générés sous forme de flux continu qu'il est inenvisageable de balayer plusieurs fois. Or, les utilisateurs de telles applications dynamiques ont besoin d'une vision plus macroscopique des données, d'analyser les tendances générales et leurs variations au cours du temps. Calculer des cubes à partir de flots de données s'avère donc très pertinent.

Des travaux de recherche ont proposé différentes variations autour du concept de cube de données. Par exemple, les cubes icebergs (Beyer *et al.*, 1999) sont des cubes de données partiels qui, à l'instar des motifs fréquents, ne recèlent que les tendances suffisamment générales pour être pertinentes, en imposant aux valeurs des différentes mesures d'être supérieures à des seuils minimaux donnés. Les cubes intervallaires (Casali *et al.*, 2003c) peuvent être vus comme une extension des cubes icebergs dans la mesure où ils permettent à l'utilisateur de se focaliser sur les tendances qui s'inscrivent dans une « *fenêtre* » significative (*i.e.* les mesures sont bornées par des seuils minimaux et maximaux). Enfin, les nouvelles tendances apparaissant (ou les tendances avérées disparaissant) lors du rafraîchissement d'un entrepôt ou dans un flot de données sont mises en évidence par le calcul de cubes différentiels (Casali, 2004). Ceux-ci peuvent être perçus comme la différence ensembliste entre deux cubes : l'un stocké dans l'entrepôt et l'autre calculé à partir des données de rafraîchissement. Suivant l'ordre des deux opérandes, sont exhibées les tendances disparaissantes ou apparaissantes.

Souvent ces différents types de cubes, à commencer par le cube de données original, n'ont pas été appréhendés comme des concepts mais comme le résultat de requêtes ou d'algorithmes plus efficaces.

Dans cet article, nous proposons une nouvelle structure unificatrice qui nous permet de caractériser les différents cubes évoqués et nous définissons une nouvelle variante : les cubes émergents. Plus précisément, nos contributions sont les suivantes :

(i) nous établissons les fondements d'une nouvelle structure appelée cube convexe qui s'appuie sur l'espace de recherche du treillis cube. Le cube convexe prend en compte des combinaisons de contraintes monotones et anti-monotones. Nous montrons que cette structure est un espace convexe (Vel, 1993) et qu'elle peut donc être représentée par ses bordures ;

(ii) grâce à la structure de cube convexe, nous introduisons les définitions formelles des cubes de données, cubes icebergs, cubes intervallaires et cubes différentiels ;

(iii) enfin, en nous inspirant de (Dong *et al.*, 2005) qui définit les motifs émergents dans un contexte binaire et pour la classification supervisée, nous proposons le concept

de cube émergent. Celui-ci capture des tendances non significatives mais qui, lors d'un rafraîchissement, le deviennent dans des proportions pertinentes pour l'utilisateur. Il permet aussi, de manière symétrique, d'exhiber des tendances significatives qui chutent au point de ne plus l'être. Outre les tendances apparaissant ou disparaissant (capturées par le cube différentiel), le cube émergent permet au décideur de connaître et d'analyser des renversements de tendances. La mise en évidence de changements de tendances est tout aussi intéressante dans l'analyse de flots de données que dans le contexte de bases OLAP classiques mais elle est plus critique à établir puisqu'en temps réel.

Le reste de l'article est organisé de la manière suivante. Le paragraphe 2 présente notre cadre de travail en décrivant brièvement l'espace de recherche multidimensionnel que nous utilisons ensuite : le treillis cube. Au paragraphe 3, nous détaillons la structure de cube convexe. Son utilisation pour caractériser les différents types de cubes est proposée dans le paragraphe suivant. Enfin, nous introduisons le concept de cube émergent et le définissons de manière cohérente avec les précédents.

## 2. Concepts de base

Dans ce paragraphe, nous présentons le concept de treillis cube (Casali *et al.*, 2003a) permettant de formaliser les nouvelles structures introduites par notre proposition.

Tout au long de cet article, nous faisons les hypothèses suivantes et utilisons les notations introduites. Soit  $r$  une relation de schéma  $\mathcal{R}$ . Les attributs de  $\mathcal{R}$  sont divisés en deux ensembles (i)  $\mathcal{D}$  l'ensemble des attributs dimensions (aussi appelés catégories ou attributs nominaux) qui correspondent aux critères d'analyse et (ii)  $\mathcal{M}$  l'ensemble des attributs mesures.

### 2.1. Espace de recherche : treillis cubes

L'espace multidimensionnel d'une relation d'attributs catégories  $r$  regroupe toutes les combinaisons valides construites en considérant l'ensemble des valeurs des attributs de  $\mathcal{D}$ , ensemble enrichi de la valeur symbolique ALL.

L'espace multidimensionnel de  $r$  est noté et défini comme suit :  $Space(r) = (\times_{A \in \mathcal{D}} (Dim(A) \cup ALL)) \cup \{(\emptyset, \dots, \emptyset)\}$  où  $\times$  symbolise le Produit cartésien,  $(\emptyset, \dots, \emptyset)$ , le majorant universel et  $Dim(A)$  la projection de  $r$  sur  $A$ . Toute combinaison de  $Space(r)$  est un tuple et représente un motif multidimensionnel.

L'espace multidimensionnel de  $r$  est structuré par la relation de généralisation/spécialisation entre tuples, notée  $\preceq_g$ . Cet ordre est originellement introduit par T. Mitchell (Mitchell, 1982) dans le cadre de l'apprentissage de concepts. Dans un contexte de gestion de cubes de données, cet ordre a la même sémantique que celle des opérateurs ROLLUP/DRILLDOWN sur le cube (Gray *et al.*, 1997) et sert de comparateur entre les

tuples (cellules) du cube quotient (Lakshmanan *et al.*, 2002). Soit  $u, v$  deux tuples de l'espace multidimensionnel de  $r$  :

$$u \preceq_g v \Leftrightarrow \begin{cases} \forall A \in \mathcal{D} \text{ tel que } u[A] \neq \text{ALL}, \\ u[A] = v[A] \\ \text{ou } v = (\emptyset, \dots, \emptyset) \end{cases}$$

Si  $u \preceq_g v$ , nous disons que  $u$  est plus général que  $v$  dans  $Space(r)$ .

*Exemple 1* - Considérons la relation  $\text{DOCUMENT}_1$  (cf table 1) répertoriant les quantités vendues par Type, par Ville et par Éditeur. Dans l'espace multidimensionnel de cette relation, nous avons :  $(\text{Roman}, \text{ALL}, \text{ALL}) \preceq_g (\text{Roman}, \text{Marseille}, \text{Gallimard})$ , c'est-à-dire que le tuple  $(\text{Roman}, \text{ALL}, \text{ALL})$  est plus général que  $(\text{Roman}, \text{Marseille}, \text{Gallimard})$  et  $(\text{Roman}, \text{Marseille}, \text{Gallimard})$  est plus spécifique que  $(\text{Roman}, \text{ALL}, \text{ALL})$ . De plus, tout motif multidimensionnel généralise le tuple  $(\emptyset, \emptyset, \emptyset)$  et spécialise le tuple  $(\text{ALL}, \text{ALL}, \text{ALL})$ .

Type	Ville	Éditeur	Quantité
Roman	Marseille	Gallimard	2
Roman	Marseille	Hachette	2
Scolaire	Paris	Hachette	1
Essai	Paris	Hachette	6
Scolaire	Marseille	Hachette	1

**Tableau 1.** Relation exemple  $\text{DOCUMENT}_1$

Les deux opérateurs de base définis pour la construction de tuples sont la Somme (notée  $+$ ) et le Produit (noté  $\bullet$ ). La somme de deux tuples retourne le tuple le plus spécifique généralisant les deux opérandes. Soit  $u$  et  $v$  deux tuples de  $Space(r)$ ,  $t = u + v \Leftrightarrow \forall A \in \mathcal{D}$ ,

$$t[A] = \begin{cases} u[A] \text{ si } u[A] = v[A] \\ \text{ALL sinon.} \end{cases}$$

Nous disons que  $t$  est la somme des tuples  $u$  et  $v$ .

*Exemple 2* - Dans notre exemple, nous avons  $(\text{Roman}, \text{Marseille}, \text{Gallimard}) + (\text{Roman}, \text{Marseille}, \text{Hachette}) = (\text{Roman}, \text{Marseille}, \text{ALL})$ . Ceci implique que  $(\text{Roman}, \text{Marseille}, \text{ALL})$  est construit à partir des tuples  $(\text{Roman}, \text{Marseille}, \text{Gallimard})$  et  $(\text{Roman}, \text{Marseille}, \text{Hachette})$ .

Le produit de deux tuples retourne le tuple le plus général spécialisant les deux opérandes. Si pour ces deux tuples, il existe un attribut  $A$  prenant des valeurs distinctes et réelles (i.e. existant dans la relation initiale), alors seul le tuple  $(\emptyset, \dots, \emptyset)$  les

spécialise (hormis ce tuple, les ensembles permettant de les construire sont disjoints). Soit  $u$  et  $v$  deux tuples de  $Space(r)$ , alors :  $t = u \bullet v \Leftrightarrow$

$$\begin{cases} t = (\emptyset, \dots, \emptyset) \text{ si } \exists A \in \mathcal{D} \text{ tel que } u[A] \neq v[A] \neq \text{ALL}, \\ \text{sinon } \forall A \in \mathcal{D} \begin{cases} t[A] = u[A] \text{ si } v[A] = \text{ALL} \\ t[A] = v[A] \text{ si } u[A] = \text{ALL}. \end{cases} \end{cases}$$

Nous disons que  $t$  est le produit des tuples  $u$  et  $v$ .

*Exemple 3* - Nous avons  $(\text{Roman}, \text{ALL}, \text{ALL}) \bullet (\text{ALL}, \text{Marseille}, \text{ALL}) = (\text{Roman}, \text{Marseille}, \text{ALL})$ . Ainsi,  $(\text{Roman}, \text{ALL}, \text{ALL})$  et  $(\text{ALL}, \text{Marseille}, \text{ALL})$  généralisent  $(\text{Roman}, \text{Marseille}, \text{ALL})$  et ce dernier tuple participe à la construction de  $(\text{Roman}, \text{ALL}, \text{ALL})$  et de  $(\text{ALL}, \text{Marseille}, \text{ALL})$  (directement ou non). Les tuples  $(\text{Roman}, \text{ALL}, \text{ALL})$  et  $(\text{Scolaire}, \text{ALL}, \text{ALL})$  n'ont d'autre point commun que le tuple de valeurs vides (i.e. le tuple  $(\emptyset, \emptyset, \emptyset)$ ).

En dotant l'espace multidimensionnel  $Space(r)$  de la relation de généralisation entre tuples et en utilisant les opérateurs Produit et Somme, nous introduisons une structure algébrique appelée treillis cube qui fixe un cadre théorique et général pour l'OLAP et la fouille de bases de données multidimensionnelles.

**Theorème 2.1** - Soit  $r$  une relation d'attributs catégories. L'ensemble ordonné  $CL(r) = \langle Space(r), \preceq_g \rangle$  est un treillis complet appelé treillis cube dans lequel les opérateurs Meet ( $\wedge$ ) et Join ( $\vee$ ) sont définis par :

- 1)  $\forall T \subseteq CL(r), \bigwedge T = +_{t \in T} t$
- 2)  $\forall T \subseteq CL(r), \bigvee T = \bullet_{t \in T} t$

### 3. Cubes convexes

Dans ce paragraphe, nous étudions la structure du treillis cube en présence de conjonctions de contraintes monotones et/ou antimonotones selon la généralisation. Nous montrons que cette structure est un espace convexe qu'on appelle cube convexe. Nous proposons des représentations condensées (avec bordures) du cube convexe avec un double objectif : définir d'une manière compacte l'espace de solutions et décider si un tuple  $t$  appartient, ou pas, à cet espace.

Nous prenons en compte les contraintes monotones et/ou antimonotones les plus couramment utilisées en fouille de base de données (Pei *et al.*, 2002). Celles-ci peuvent porter sur :

- des mesures d'intérêts comme la fréquence de motifs, la confiance, la corrélation (Han *et al.*, 2001) : dans ce cas, seuls les attributs dimensions de  $\mathcal{R}$  sont nécessaires ;
- des agrégats selon des attributs mesures  $\mathcal{M}$  calculés en utilisant des fonctions statistiques additives (COUNT, SUM, MIN, MAX).

Nous rappelons les définitions de la notion d'espace convexe, des contraintes monotones et antimonotones selon l'ordre de généralisation  $\preceq_g$ .

**Définition 1 [Espace Convexe]** - Soit  $(\mathcal{P}, \leq)$  un ensemble partiellement ordonné,  $\mathcal{C} \subseteq \mathcal{P}$  est un espace convexe (Vel, 1993) si et seulement si  $\forall x, y, z \in \mathcal{P}$  tel que  $x \leq y \leq z$  et  $x, z \in \mathcal{C} \Rightarrow y \in \mathcal{C}$ .

Donc  $\mathcal{C}$  est borné par deux ensembles : (i) un majorant (ou « Upper set »), noté  $S$ , défini par  $S = \max_{\leq}(\mathcal{C})$ , (ii) un minorant (ou « Lower set »), noté  $G$  et défini par  $G = \min_{\leq}(\mathcal{C})$ .

**Définition 2 [Contraintes monotones/antimonotones selon la généralisation]** -

1) Une contrainte  $Const$  est dite monotone pour l'ordre de généralisation si et seulement si :  $\forall t, u \in CL(r) : [t \preceq_g u \text{ et } Const(t)] \Rightarrow Const(u)$ .

2) Une contrainte  $Const$  est dite antimonotone pour l'ordre de généralisation si et seulement si :  $\forall t, u \in CL(r) : [t \preceq_g u \text{ et } Const(u)] \Rightarrow Const(t)$ .

*Notations* : nous notons  $cmc$  (respectivement  $came$ ) une conjonction de contraintes monotones (respectivement antimonotones) et  $chc$  une conjonction hybride de contraintes (monotones et antimonotones). En reprenant les symboles  $S$  et  $G$  introduits dans (Mitchell, 1982) et suivant les cas considérés, les bornes introduites sont indicées par le type de contrainte considérée. Par exemple  $S_{came}$  symbolise l'ensemble des tuples les plus spécifiques vérifiant la conjonction de contraintes antimonotones.

**Remarque 1** - Pour éviter les ambiguïtés faites en apprentissage (Raedt *et al.*, 2001), il est important de noter que les bornes  $S_{chc}$  et  $G_{chc}$  ne sont pas les mêmes que les ensembles  $S$  et  $G$  définis dans le cadre de l'espace de versions, car un espace de versions est un espace convexe, mais tout espace convexe n'est pas un espace de versions, à cause des contraintes considérées.

**Exemple 4** - Dans l'espace multidimensionnel exemple de la relation  $DOCUMENT_1$  (cf. tableau 1), nous voulons connaître tous les tuples dont la somme des valeurs pour l'attribut mesure *Quantité* est supérieure ou égale à 3. La contrainte «  $SUM(Quantité) \geq 3$  » est une contrainte antimonotone. Si le total des ventes par Type, par Ville et par Éditeur est supérieur à 3, il l'est *a fortiori* pour un niveau plus agrégé de granularité *e.g.* par Type et par Éditeur (toutes villes confondues) ou par Ville (tous types et éditeurs confondus). De même, si nous voulons connaître tous les tuples dont la somme des valeurs pour l'attribut *Quantité* est inférieure ou égale à 6, la contrainte exprimée «  $SUM(Quantité) \leq 6$  » est monotone. Considérons que le total des ventes par type (les attributs Ville et Éditeur ont pour valeur ALL) respecte cette contrainte, la même information observée à un niveau de détail plus fin satisfait forcément la même condition et donc la somme des ventes par Type et par Ville est inférieure à 6 de même que la somme des ventes par Type, par Ville et par Éditeur.

**Remarque 2** – Nous supposons par la suite que le tuple  $(ALL, \dots, ALL)$  vérifie toujours la conjonction de contraintes antimonotones et que le tuple  $(\emptyset, \dots, \emptyset)$  vérifie

toujours la conjonction de contraintes monotones. Avec ces hypothèses, l'espace des solutions contient au moins un élément (éventuellement le tuple de valeurs vides).

– De plus, nous supposons que le tuple  $(ALL, \dots, ALL)$  ne vérifie jamais la conjonction de contraintes monotones et que le tuple  $(\emptyset, \dots, \emptyset)$  ne vérifie jamais la conjonction de contraintes antimonotones, car sinon, l'espace de solutions est  $Space(r)$ .

**Theorème 3.1** - Tout treillis cube avec contraintes monotones et/ou antimonotones est un espace convexe qu'on appelle cube convexe,  $CC(r)_{const} = \{t \in CL(r) \mid const(t)\}$ , où  $const$  peut être  $cmc$ ,  $camc$  ou  $chc$  suivant que la combinaison de contraintes est monotone, antimonotone ou hybride. Son majorant  $S_{const}$  et son minorant  $G_{const}$  sont :

$$\begin{aligned} 1. \text{ si } const = cmc, & \begin{cases} G_{cmc} = \min_{\preceq_g}(CC(r)_{cmc}) \\ S_{cmc} = (\emptyset, \dots, \emptyset) \end{cases} \\ 2. \text{ si } const = camc, & \begin{cases} G_{camc} = (ALL, \dots, ALL) \\ S_{camc} = \max_{\preceq_g}(CC(r)_{camc}) \end{cases} \\ 3. \text{ si } const = chc, & \begin{cases} G_{chc} = \min_{\preceq_g}(CC(r)_{chc}) \\ S_{chc} = \max_{\preceq_g}(CC(r)_{chc}) \end{cases} \end{aligned}$$

**Preuve 3.1** 1) Soit  $CC(r)_{cmc} = \{t \in CL(r) \mid \exists u \in S_{cmc} \text{ et } \exists v \in G_{cmc} : t \preceq_g u \text{ et } v \preceq_g t\}$ . Nous montrons que  $CC(r)_{cmc}$  est l'ensemble des tuples satisfaisant la conjonction de contraintes monotones. Pour les besoins de la démonstration, notons  $Sol_{cmc}$  cet ensemble solution. Soit  $t \in CC(r)_{cmc}$ .

- Par définition, il existe  $v \in G_{cmc} \mid cmc(v)$  et  $v \preceq_g t$ . Puisque  $cmc$  est une contrainte monotone, nous avons  $cmc(t)$ . Par conséquent  $t \in Sol_{cmc}$ . Donc  $CC(r)_{cmc} \subseteq Sol_{cmc}$ . (a)

- Soit  $t \in Sol_{cmc}$ , il existe forcément  $v \in G_{cmc} \mid v \preceq_g t$  car  $G_{cmc}$  représente les minimaux vérifiant  $cmc$ . De plus la contrainte  $\exists u \in S_{cmc} \mid t \preceq_g u$  est toujours vérifiée. Donc  $t \in CC(r)_{cmc}$  et  $Sol_{cmc} \subseteq CC(r)_{cmc}$ . (b)

(a) et (b)  $\Rightarrow Sol_{cmc} = CC(r)_{cmc}$

2) vrai par application du principe de dualité (Ganter *et al.*, 1999) sur le cube convexe avec une conjonction de contraintes monotones.

3) vrai car si  $Sol_{chc} = CC(r)_{chc}$ , alors  $Sol_{chc} = Sol_{cmc} \cap Sol_{camc}$ . L'application des deux caractérisations précédentes permet de déduire celle pour  $chc$ .

Le majorant  $S_{const}$  représente les tuples les plus spécifiques satisfaisant la conjonction de contraintes et le minorant  $G_{const}$  les tuples les plus généraux satisfaisant la conjonction de contraintes. Donc  $S_{const}$  et  $G_{const}$  permettent d'obtenir des représentations condensées du cube convexe en présence d'une conjonction de contraintes monotones et/ou antimonotones.

Le corollaire suivant permet la caractérisation des bordures du cube convexe en présence d'une conjonction hybride de contraintes  $chc = camc \wedge cmc$  en ne connaissant que (i) soit la bordure maximale pour la contrainte antimonotone ( $S_{camc}$ ) et la contrainte monotone  $cmc$ , (ii) soit la bordure minimale pour la contrainte monotone ( $G_{cmc}$ ) et la contrainte antimonotone  $S_{camc}$ .

### Corollaire 3.1

1) Étant donné  $S_{camc}$  et  $cmc$ , les bordures de l'ensemble du cube convexe  $CC(r)_{chc}$  :

$$\begin{cases} G_{chc} = \min_{\preceq_g} (\{t \in CL(r) \mid \exists t' \in S_{camc} : \\ t \preceq_g t' \text{ et } cmc(t)\}) \\ S_{chc} = \{t \in S_{camc} \mid \exists t' \in G_{chc} : t' \preceq_g t\} \end{cases}$$

2) Étant donné  $G_{cmc}$  et  $camc$ , une représentation condensée de  $CC(r)_{chc}$  est :

$$\begin{cases} S_{chc} = \max_{\preceq_g} (\{t \in CL(r) \mid \exists t' \in G_{cmc} : \\ t' \preceq_g t \text{ et } camc(t)\}) \\ G_{chc} = \{t \in G_{cmc} \mid \exists t' \in S_{chc} : t \preceq_g t'\}. \end{cases}$$

*Exemple 5* - Le tableau 2 donne les bornes  $S_{camc}$ ,  $S_{chc}$ ,  $G_{cmc}$  et  $G_{chc}$  du cube convexe de la relation exemple en considérant la contrainte hybride «  $3 \leq \text{SUM}(\text{Quantité}) \leq 6$  ».

La caractérisation du cube convexe comme un espace convexe nous permet de savoir, en ne connaissant que les bordures du cube convexe, si un tuple quelconque satisfait ou pas la conjonction de contraintes. En effet, si un tuple de  $Space(r)$  vérifie une conjonction de contraintes antimonotones alors tout tuple le généralisant la satisfait aussi. Dualement, si un tuple vérifie une conjonction de contraintes monotones, alors tous les tuples le spécialisant satisfont aussi ces contraintes. La représentation par bordure du cube convexe de la relation  $\text{DOCUMENT}_1$  (cf. tableau 2) permet de répondre facilement à des requêtes telles que :

- 1) Est ce que le nombre d'achats à Marseille est compris entre 3 et 6 ?
- 2) Est ce que le nombre de livres scolaires vendus à Paris est compris entre 3 et 6 ?
- 3) Est ce que le nombre de romans vendus à Aubagne par les éditions Hachette est compris entre 3 et 6 ?

La réponse à la première question est oui car le tuple (ALL, Marseille, ALL), donnant les achats effectués dans la ville de Marseille tous types et éditeurs confondus, appartient à la bordure  $G_{chc}$ . Il en est de même pour la seconde requête (nombre de livres scolaires vendus à Paris) car le tuple (Scolaire, Paris, ALL) spécialise le tuple (Scolaire, ALL, ALL) appartenant à  $G_{chc}$  et généralise le tuple (Scolaire, Paris, Hachette) appartenant à la bordure  $S_{chc}$ . En revanche, la réponse à la troisième question est non car le tuple (ALL, Paris, Hachette) (tous les livres édités

par Hachette achetés à Paris) ne spécialise aucun tuple de la bordure  $G_{chc}$  et ce même s'il généralise le tuple (Scolaire, Paris, Hachette) de la bordure  $S_{chc}$ .

$S_{camc}$	(Roman, Marseille, ALL) (ALL, Marseille, Hachette) (Scolaire, Paris, Hachette)
$S_{chc}$	(Roman, Marseille, ALL) (ALL, Marseille, Hachette) (Scolaire, Paris, Hachette)
$G_{cmc}$	(Roman, ALL, ALL) (Essai, ALL, ALL) (Scolaire, ALL, ALL) (ALL, Marseille, ALL) (ALL, ALL, Gallimard)
$G_{chc}$	(Roman, ALL, ALL) (Scolaire, ALL, ALL) (ALL, Marseille, ALL)

**Tableau 2.** Bornes du cube convexe pour «  $3 \leq \text{SUM}(\text{Quantité}) \leq 6$  »

#### 4. Formalisation de cubes existants

Dans ce paragraphe, nous passons en revue différentes variantes des cubes de données et, en utilisant la structure de cube convexe, nous en proposons une caractérisation à la fois solide et simple.

##### 4.1. Cubes de données

Originellement proposé dans (Gray *et al.*, 1997), le cube de données selon un ensemble de dimensions est présenté comme le résultat de tous les GROUP BY qu'il est possible de formuler selon une combinaison de ces dimensions. Le résultat de chaque GROUP BY est appelé un cuboïde et l'ensemble de tous les cuboïdes est structuré au sein d'une relation notée  $\text{Datacube}(r)$ . Le schéma de cette relation reste le même que celui de  $r$ , à savoir  $\mathcal{D} \cup \mathcal{M}$  et c'est ce même schéma qui est utilisé pour tous les cuboïdes (afin de pouvoir en faire l'union) en mettant en œuvre une idée simple : toute dimension ne participant pas au calcul d'un cuboïde (i.e. ne figurant pas dans la clause GROUP BY) se voit attribuer la valeur ALL.

Pour tout ensemble d'attributs  $X \subseteq \mathcal{D}$ , un cuboïde du cube de données, noté  $\text{Cuboid}(X, f(\{\mathcal{M}|\ast\}))$ , peut être obtenu comme suit en utilisant une requête SQL :

```
SELECT [ALL,] X, f({M|*})
```

```
FROM r
GROUP BY X;
```

Ainsi, nous obtenons deux requêtes SQL pour exprimer le calcul d'un cube de données :

1) soit en utilisant l'opérateur GROUP BY CUBE (ou CUBE BY selon le SGBD) :

```
SELECT D, f({M|*})
FROM r
GROUP BY CUBE (D);
```

2) soit en faisant l'union de tous les cuboïdes :

$Datacube(r, f(\{M|*\})) = \bigcup_{X \subseteq \mathcal{D}} Cuboid(X, f(\{M|*\}))$ . Cette requête s'exprime

comme suit en SQL :

```
SELECT ALL, ..., f({M|*})
FROM r
UNION
SELECT A, ALL, ..., f({M|*})
FROM r
GROUP BY A
UNION
...
```

*Exemple 6* - Dans notre exemple, l'ensemble de toutes les requêtes agrégatives peut être exprimé en utilisant l'opérateur GROUP BY CUBE comme suit :

```
SELECT Type, Ville, Éditeur, SUM(Quantite)
FROM Document1
GROUP BY CUBE Type, Ville, Éditeur;
```

Cette requête a pour résultat le calcul de  $2^3 = 8$  cuboïdes :  $TVE, TE, TV, VE, T, V, E$  et  $\emptyset$  (en considérant les initiales des attributs). Le cuboïde selon  $TVE$  correspond à la relation initiale elle-même.

Un tuple  $t$  appartient au cube de données d'une relation  $r$  si et seulement s'il existe au moins un tuple  $t'$  de  $r$  qui spécialise  $t$ ; sinon  $t$  ne peut pas être construit. Par conséquent, quelle que soit la fonction agrégative, les tuples constituant le cube de données restent invariants, seules les valeurs calculées par la fonction agrégative changent.

**Proposition 4.1** - Soit  $r$  une relation projetée sur  $\mathcal{D}$ , l'ensemble des tuples (*i.e.* hormis les valeurs des attributs mesures) constituant le cube de données de  $r$  est un cube convexe pour la contrainte « COUNT(\*)  $\geq 1$  » :

$$Datacube(r) = \{t \in CL(r) \mid t[Count(*)] \geq 1\}$$

**Preuve 4.1** - D'après la définition d'un cube de données :  $t \in \text{datacube}(r) \Leftrightarrow \exists t' \in r \mid t \preceq_g t' \Leftrightarrow t[\text{COUNT}(*)] \geq 1$  d'après la définition de la fonction COUNT.

Puisque la contrainte «  $\text{COUNT}(* ) \geq 1$  » est une contrainte antimonotone (selon  $\preceq_g$ ), un cube de données est un cube convexe. En appliquant le théorème 3.1, nous déduisons que tout cube de données peut être représenté par deux bordures : la relation  $r$  qui est le majorant et le tuple  $(\text{ALL}, \dots, \text{ALL})$  qui est le minorant. Ainsi, nous pouvons facilement tester l'appartenance d'un tuple quelconque  $t$  au cube de données de  $r$  : il suffit de trouver un tuple  $t' \in r$  qui spécialise  $t$ .

*Exemple 7* - Avec la relation exemple  $\text{DOCUMENT}_1$  (cf. tableau 1), le tuple (Roman, Marseille, ALL) appartient au cube de données car il est spécialisé par le tuple (Roman, Marseille, Gallimard) de la relation initiale.

Dans les sous-paragraphes suivants, en nous appuyant toujours sur la structure de cube convexe, nous proposons une caractérisation des différentes déclinaisons du cube de données.

#### 4.2. Cubes icebergs

En s'inspirant des motifs fréquents, (Beyer *et al.*, 1999) introduit les cubes icebergs qui sont présentés comme des sous-ensembles de tuples du cube de données satisfaisant, pour les valeurs de la mesure, une contrainte de seuil minimum. L'objectif sous-jacent est triple. Il s'agit dans ce cas d'exhiber les tendances suffisamment générales pour être pertinentes pour le décideur, il en découle deux intérêts techniques importants : ne pas calculer ni matérialiser la totalité du cube d'où un gain notable à la fois de temps d'exécution et d'espace disque. La requête SQL permettant de calculer un cube iceberg prend la forme suivante :

```
SELECT D, f({M|*})
FROM r
GROUP BY CUBE D
HAVING f({M|*}) >= MinSeuil;
```

En nous appuyant sur la définition du cube convexe, nous formalisons le concept de cube iceberg dans la proposition suivante :

**Proposition 4.2** - La contrainte «  $f(\{M|*\}) \geq \text{MinSeuil}$  » étant une contrainte antimonotone (Pei *et al.*, 2002), le cube iceberg est un cube convexe caractérisé comme suit :

$$\text{CubeIceberg}(r) = \{t \in CL(r) \mid t[f(\{M|*\})] \geq \text{MinSeuil}\}.$$

*Exemple 8* - Avec la relation exemple  $\text{DOCUMENT}_1$  (cf. tableau 1), la bordure  $S$  relative à la contrainte «  $\text{SUM}(\text{Quantité}) \geq 3$  » est composée des trois tuples suivants :

{ (Roman, Marseille, ALL), (ALL, Marseille, Hachette), (Roman, Paris, Hachette) }.  
La bordure  $G$  est uniquement composée du tuple ne contenant que des valeurs ALL.

### 4.3. Cubes intervallaires

Le cube intervallaire ne contient que les tuples du cube de données pour lesquels les valeurs de la mesure sont comprises dans un intervalle donné. La requête SQL permettant de calculer un tel cube est la suivante :

```
SELECT D, f({M|*})
FROM r
GROUP BY CUBE D
HAVING f({M|*}) BETWEEN MinSeuil AND MaxSeuil;
```

Dans le cadre de travail établi, la caractérisation du cube intervallaire est la suivante :

**Proposition 4.3** - La contrainte «  $f(\{M|*\}) \geq \text{MinSeuil}$  » étant une contrainte antimotone (Pei *et al.*, 2002) et la contrainte «  $f(\{M|*\}) \leq \text{MaxSeuil}$  » étant une contrainte monotone, le cube intervallaire est un cube convexe. Ainsi, nous avons :

$$\text{CubeIntervalaire}(r) = \{t \in CL(r) \mid \text{MaxSeuil} \geq t[f(\{M|*\})] \geq \text{MinSeuil}\}.$$

*Exemple 9* - Avec la relation exemple DOCUMENT<sub>1</sub> (cf. tableau 1), les bordures  $S$  et  $G$  relatives à la contrainte «  $\text{SUM}(\text{Quantité}) \in [3, 6]$  » sont données dans le tableau 3.

$S$	(Roman, Marseille, ALL) (ALL, Marseille, Hachette) (Scolaire, Paris, Hachette)
$G$	(Roman, ALL, ALL) (Scolaire, ALL, ALL) (ALL, Marseille, ALL)

**Tableau 3.** Bornes du cube intervallaire pour la contrainte «  $\text{SUM}(\text{Quantité}) \in [3, 6]$  »

### 4.4. Cubes différentiels

Les cubes différentiels sont le résultat de la différence entre les cubes de deux relations  $r_1$  et  $r_2$ . Ils mettent en évidence des tuples pertinents dans un des cubes et inexistant dans l'autre. Leur intérêt est donc de pouvoir comparer des tendances

entre deux jeux de données. Par exemple, dans une application distribuée où deux relations rassemblent des données collectées dans des zones géographiques différentes, le cube différentiel exhibe des tendances significatives dans une zone mais inexistantes dans l'autre. Si l'on considère non plus la dimension géographique mais la dimension temporelle, les cubes différentiels permettent d'isoler des tendances fréquentes à un instant et qui disparaissent ou des tendances inexistantes qui apparaissent de manière significative. Considérons que la relation originelle est  $r_1$  et que les tuples alimentant l'entrepôt lors d'un rafraîchissement sont stockés dans  $r_2$ , le cube différentiel peut être obtenu par la requête SQL suivante :

```
SELECT D, f({M|*})
  FROM r2
  GROUP BY CUBE D
  HAVING f({M|*}) >= MinSeuil
MINUS
SELECT D, f({M|*})
  FROM r1
  GROUP BY CUBE D;
```

De manière cohérente avec les types de cubes précédents, nous proposons une caractérisation des cubes différentiels.

**Proposition 4.4** - La contrainte «  $f(\{M|*\}) \geq \text{MinSeuil}$  » étant une contrainte antimonotone (Pei *et al.*, 2002) et la contrainte « *n'appartient pas au cube de données de la relation  $r_1$*  » étant une contrainte monotone (*cf.* paragraphe 4.1 et application du principe de dualité (Ganter *et al.*, 1999)), le cube différentiel est un cube convexe. Ainsi, nous avons :

$$\text{CubeDifferentiel}(r_2, r_1) = \{t \in CL(r_1 \cup r_2) \mid t[f(\{M|*\})] \geq \text{MinSeuil} \\ \text{et } \nexists t' \in r_2 \mid t \preceq_g t'\}.$$

*Exemple 10* - Soit la relation  $\text{DOCUMENT}_2$  suivante :

RowId	Type	Ville	Éditeur	Quantité
1	Scolaire	Marseille	Gallimard	3
2	Scolaire	Paris	Hachette	3
3	Scolaire	Marseille	Hachette	1
4	Roman	Marseille	Gallimard	3
5	Essai	Paris	Hachette	2
6	Essai	Paris	Gallimard	2
7	Essai	Marseille	Hachette	1

**Tableau 4.** Relation exemple  $\text{DOCUMENT}_2$

Les bordures  $S$  et  $G$  du cube différentiel entre les relations  $\text{DOCUMENT}_2$  et  $\text{DOCUMENT}_1$ , pour le seuil  $\text{MinSeuil} = 1/15$  et la fonction agrégative SUM, sont données dans le tableau 5.

$S$	(Scolaire, Marseille, Gallimard)
	(Scolaire, Marseille, Hachette)
	(Essai, Paris, Gallimard)
$G$	(Essai, ALL, Gallimard)
	(Scolaire, Marseille, ALL)
	(Scolaire, ALL, Gallimard)
	(ALL, Paris, Galimard)

**Tableau 5.** Bornes du cube différentiel

**Remarque 3** - Dans un souci d'homogénéité, nous terminons ce paragraphe en donnant la requête SQL générique qui correspond au calcul du cube convexe :

```
SELECT D, f({M|*})
FROM r
GROUP BY CUBE D
[HAVING condition(s) anti-monotone(s)
AND Condition(s) monotone(s)];
```

## 5. Cubes émergents

Dans ce paragraphe, nous introduisons le concept de cube émergent. De tels cubes exhibent des tendances non pertinentes pour l'utilisateur (parce qu'en deçà d'un seuil) mais qui le deviennent ou au contraire des tendances significatives qui s'atténuent sans forcément disparaître. Les cubes émergents permettent donc d'élargir les résultats des cubes différentiels en affinant les comparaisons entre deux cubes. Ils sont tout aussi intéressants dans un contexte OLAP que pour l'analyse de flots de données car ils mettent en évidence des renversements de tendances.

Dans la suite, nous considérons uniquement les fonctions agrégatives COUNT et SUM. Pour conserver la propriété d'antimonotonie de SUM, nous supposons que toutes les valeurs prises par la mesure sont strictement positives et introduisons une version relative de ces fonctions.

**Définition 3 [Fonction agrégative relative]** - Soit  $r$  une relation,  $t \in CL(r)$  un tuple, et  $f \in \{\text{SUM, COUNT}\}$  une fonction agrégative. On appelle  $f_{rel}(\cdot, r)$  la fonction agrégative relative de la fonction  $f$  pour la relation  $r$ .  $f_{rel}(t, r)$  est le ratio entre la

valeur de  $f$  pour le tuple  $t$  et la valeur de  $f$  appliquée sur toute la relation  $r$  (donc pour le tuple  $(ALL, \dots, ALL)$ ).

$$f_{rel}(t, r) = \frac{f(t, r)}{f((ALL, \dots, ALL), r)}$$

Par exemple, la fonction  $COUNT_{rel}(t, r)$  correspond simplement à  $Freq(t, r)$  (la fréquence d'apparition du motif multidimensionnel  $t$  dans la relation  $r$ ).

**Remarque 4** - Étant donné que  $f$  est additive et que les valeurs prises par la mesure sont strictement positives, on a  $0 < f(t, r) < f((ALL, \dots, ALL), r)$  et par conséquent  $0 < f_{rel}(t, r) < 1$

Disposant de deux jeux de données unicompatibles  $r_1$  et  $r_2$ , nous nous intéressons aux tendances « non significatives » dans  $r_1$  mais qui le deviennent dans  $r_2$  dans des proportions pertinentes pour l'utilisateur. Les entrepôts aussi bien que les flots de données incluant nécessairement la dimension chronologique,  $r_1$  et  $r_2$  peuvent être typiquement vus comme des ensembles de tuples pourvus d'estampilles temporelles différentes. Ainsi,  $r_1$  peut déjà être stockée dans l'entrepôt et  $r_2$  rassembler les nouveaux tuples à insérer lors d'un rafraîchissement.  $r_1$  et  $r_2$  peuvent aussi correspondre à des ensembles de tuples collectés lors de deux intervalles de temps. Même si la dimension temps est prépondérante dans notre contexte de travail, elle n'est pas nécessairement le critère de comparaison entre  $r_1$  et  $r_2$ . Ainsi, dans une application distribuée, les tuples de  $r_1$  et  $r_2$  peuvent être collectés sur deux sites différents. Les tuples émergents de  $r_1$  vers  $r_2$  peuvent être simplement caractérisés par deux contraintes de seuil.

**Définition 4 [Tuple émergent]** - Un tuple  $t \in CL(r_2 \cup r_1)$  est dit émergent de  $r_1$  vers  $r_2$  si et seulement s'il satisfait les deux contraintes suivantes :

$$(C_1) f_{rel}(t, r_1) \leq MinSeuil_1$$

$$(C_2) f_{rel}(t, r_2) \geq MinSeuil_2$$

où  $MinSeuil_1$  et  $MinSeuil_2 \in ]0, 1[$

**Exemple 11** - Soit les seuils  $MinSeuil_1 = 1/3$  pour la relation  $DOCUMENT_1$  (cf. tableau 1) et  $MinSeuil_2 = 1/5$  relatif à la relation  $DOCUMENT_2$  (cf. tableau 4), le tuple  $t_1 = (Essai, Paris, ALL)$  est émergent de  $DOCUMENT_1$  vers  $DOCUMENT_2$  car  $SUM_{rel}(t_1, r_1) = 1/12$  et  $SUM_{rel}(t_1, r_2) = 4/15$ . Par contre, le tuple  $t_2 = (Essai, Marseille, ALL)$  ne l'est pas car  $SUM_{rel}(t_2, r_2) = 1/15$ .

**Définition 5 [Cube émergent]** - Nous appelons cube émergent l'ensemble de tous les tuples de  $CL(r_2 \cup r_1)$  émergents de  $r_1$  vers  $r_2$ .

Le cube émergent, noté  $CubeEmergent(r_2, r_1)$ , est un cube convexe avec la contrainte hybride  $(C_1 \wedge C_2)$  «  $t$  est émergent de  $r_1$  vers  $r_2$  ». Il est donc dé-

fini ainsi :  $CubeEmergent(r_2, r_1) = \{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}$ , avec  $C_1(t) = f_{rel}(t, r_1) < MinSeuil_1$  et  $C_2(t) = f_{rel}(t, r_2) \geq MinSeuil_2$ .

$CubeEmergent(r_2, r_1)$  est un cube convexe avec une conjonction de contraintes monotone ( $C_1$ ) et antimonotone ( $C_2$ ). On peut donc utiliser ses bordures (cf théorème 3.1) pour répondre efficacement à la question « un tuple  $t$  est il émergent ? ».

**Définition 6 [Taux d'émergence]** - Soit  $t \in CL(r_1 \cup r_2)$  un tuple et  $f$  une fonction additive (appliquée sur les valeurs toutes positives de la mesure). Nous notons  $TE(t)$  le taux d'émergence de  $t$  entre  $r_1$  et  $r_2$  et nous le définissons ainsi :

$$TE(t) = \begin{cases} 0 & \text{si } f_{rel}(t, r_1) = 0 \quad \text{et } f_{rel}(t, r_2) = 0 \\ \infty & \text{si } f_{rel}(t, r_1) = 0 \quad \text{et } f_{rel}(t, r_2) \neq 0 \\ \frac{f_{rel}(t, r_2)}{f_{rel}(t, r_1)} & \text{sinon} \end{cases}$$

À l'instar de la mesure de corrélation (Han *et al.*, 2001), lorsqu'un tuple a un taux d'émergence strictement supérieur à 1, il est positivement émergent, sinon il est négativement émergent.

*Exemple 12* - Le tableau 6 présente l'ensemble des tuples émergents de  $DOCUMENT_1$  vers  $DOCUMENT_2$  en considérant la mesure Quantité et les seuils  $MinSeuil_1 = 1/3$  et  $MinSeuil_2 = 1/5$ .

Type	Ville	Éditeur	TE( <i>Quantité</i> )
Scolaire	Marseille	Gallimard	$\infty$
Scolaire	Marseille	ALL	$\infty$
Scolaire	ALL	Gallimard	$\infty$
ALL	Marseille	Gallimard	2.4
ALL	ALL	Gallimard	2.4
Roman	Marseille	Gallimard	1.2
Roman	ALL	Gallimard	1.2
Essai	Paris	ALL	3.2
Essai	ALL	ALL	2

**Tableau 6.** Ensemble des tuples émergents de  $DOCUMENT_1$  vers  $DOCUMENT_2$

On observe que le taux d'émergence, quand il est supérieur à 1 (donc positivement émergent), permet de caractériser les tendances significatives dans  $r_2$  et qui ne sont pas aussi marquées dans  $r_1$ . Quand ce taux est inférieur à 1, il met en évidence les tendances immergentes, *i.e.* significatives dans  $r_1$  et peu présentes ou inexistantes dans  $r_2$ .

*Exemple 13* - Dans les deux relations données en exemple, on a  $TE(\text{Scolaire}, \text{ALL}, \text{ALL}) = 2.5$ . Évidemment, plus le taux d'émergence est élevé, plus la tendance est

forte. Ainsi, le tuple ci-avant indique un bond pour la vente de livres scolaires entre  $\text{DOCUMENT}_1$  et  $\text{DOCUMENT}_2$ .

**Proposition 5.1** - Soit  $\text{MinRatio} = \frac{\text{MinSeuil}_2}{\text{MinSeuil}_1}$ ,  $\forall t \in \text{CubeEmergent}(r_2, r_1)$ , on a  $\text{TE}(t) \geq \text{MinRatio}$ .

**Preuve 5.1** -  $f_{rel}(t, r_1) \leq \text{MinSeuil}_1 \Rightarrow \frac{1}{f_{rel}(t, r_1)} \geq \frac{1}{\text{MinSeuil}_1}$ , or  
 $f_{rel}(t, r_2) \geq \text{MinSeuil}_2$   
 $\Rightarrow \frac{f_{rel}(t, r_2)}{f_{rel}(t, r_1)} \geq \frac{\text{MinSeuil}_2}{\text{MinSeuil}_1}$   
 $\Rightarrow \text{TE}(t) \geq \text{MinRatio} \square$

Tous les tuples émergents de notre exemple, ont un taux d'émergence supérieur à  $3/5$ . Ceux qui ont un taux d'émergence strictement supérieur à 1 sont positivement émergents, les autres sont donc immergents.

Le cube émergent étant un cube convexe, il peut être représenté par ses bordures et donc sans avoir à calculer ni matérialiser les deux cubes comparés. Cette capacité est particulièrement attrayante car elle permet d'isoler les renversements de tendances extrêmement rapidement et à moindre coût.

### 5.1. Transversaux cubiques

Nous présentons le concept de transversaux cubiques (Casali *et al.*, 2003b) qui est un cas particulier des transversaux d'un hypergraphe (Berge, 1989, Eiter *et al.*, 1995, Gunopulos *et al.*, 1997) dans le contexte du treillis cube.

**Définition 7 [Transversal cubique]** - Soit  $T$  un ensemble de tuples ( $T \subseteq \text{CL}(r)$ ) et soit  $t \in T$  un tuple,  $t$  est un transversal cubique de  $T$  sur  $\text{CL}(r)$  si et seulement si  $t$  est un transversal cubique et  $\forall t' \in T$ ,  $t'$  est un transversal cubique et  $t' \preceq_g t \Rightarrow t = t'$ . Les minimaux transversaux cubiques de  $T$  sont notés  $cTr(T)$  et définis comme suit :

$$cTr(T) = \min_{\preceq_g} (\{t \in \text{CL}(r) \mid \forall t' \in r, t + t' \neq (\text{ALL}, \dots, \text{ALL})\})$$

Soit  $\mathbb{A}$  une anti-chaîne de  $\text{CL}(r)$  (tous les tuples de  $\mathbb{A}$  sont incomparables selon  $\preceq_g$ ), l'ensemble des minimaux transversaux cubiques de  $T$  peut être contraint en utilisant  $\mathbb{A}$ . La nouvelle définition associée est la suivante :

$$cTr(T, \mathbb{A}) = \{t \in cTr(r) \mid \exists u \in \mathbb{A} : t \preceq_g u\}$$

*Exemple 14* - Avec la relation exemple  $DOCUMENT_1$ , nous avons le résultat suivant :  $cTr(DOCUMENT_1) = \{ (Roman, Paris, ALL), (Essai, ALL, Gallimard), (Scolaire, Marseille, ALL), (Scolaire, ALL, Gallimard), (ALL, Paris, Gallimard) \}$ .

Dans le prochain paragraphe, nous montrons que l'on peut utiliser ce concept pour donner une nouvelle formulation des bordures du cube émergent.

## 5.2. Calcul efficace des bordures

Pour calculer les bordures de l'ensemble  $CubeEmergent(r_2, r_1)$ , nous reformulons les contraintes de manière à tirer profit d'algorithmes existants qui ont fait preuve de leur efficacité : (i) Max-Miner (Bayardo, 1998) et GenMax (Gouda *et al.*, 2001) pour le calcul des maximaux cubiques, et (ii) Trans (Eiter *et al.*, 1995), CTR (Casali *et al.*, 2003b), MCTR (Casali, 2004) et (Gunopulos *et al.*, 1997) pour le calcul des minimaux transversaux cubiques. Nous ramenons la contrainte « *t est un tuple émergent* » à la recherche de maximaux cubiques fréquents et de minimaux cubiques transversaux.

Il a été démontré que la contrainte ( $C_1$ ) est une contrainte monotone et ( $C_2$ ) est une contrainte anti-monotone pour l'ordre de généralisation.

L'ensemble des tuples émergents peut être représenté via deux bordures :  $S$  qui contient l'ensemble des tuple maximaux émergents et  $G$  englobant l'ensemble des tuples minimaux émergents.

$$\begin{cases} G = \min_{\preceq_g} (\{t \in CL(r) \mid C_1(t) \wedge C_2(t)\}) \\ S = \max_{\preceq_g} (\{t \in CL(r) \mid C_1(t) \wedge C_2(t)\}) \end{cases}$$

**Proposition 5.2** - Soit  $M_1$  et  $M_2$  les tuples maximaux fréquents des relations  $r_1$  et  $r_2$  :

$$M_1 = \max_{\preceq_g} (\{t \in CL(r_1) : f_{rel}(t, r_1) \geq MinSeuil_1\})$$

$$M_2 = \max_{\preceq_g} (\{t \in CL(r_2) : f_{rel}(t, r_2) \geq MinSeuil_2\})$$

Nous pouvons alors caractériser les bordures  $S$  et  $G$  de l'ensemble des tuples émergents comme suit :

- 1)  $G = cTr(M_1, M_2)$  sur  $CL(r_1 \cup r_2)$ ,
- 2)  $S = \{t \in M_2 : \exists u \in G : u \preceq_g t\}$ .

### Preuve 5.2

$$\begin{aligned} 1) t \in G &\Leftrightarrow t \in \min_{\preceq_g} (\{u \in CL(r_1 \cup r_2) : f_{rel}(u, r_1) \leq \\ &MinSeuil_1 \text{ et } f_{rel}(u, r_2) \geq MinSeuil_2\}) \\ \Leftrightarrow t \in \min_{\preceq_g} (\{u \in CL(r_1 \cup r_2) : f_{rel}(u, r_1) \leq MinSeuil_1\}) \text{ et } \exists v \in M_2 : t \preceq_g v \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow t \in \min_{\preceq_g}(\{u \in CL(r_1 \cup r_2) : \nexists v \in M_1 : u \preceq_g v\}) \text{ et } \exists v \in M_2 : t \preceq_g v \\
&\Leftrightarrow t \in cTr(M_1) \text{ et } \exists v \in M_2 : t \preceq_g v \\
&\Leftrightarrow t \in cTr(M_1, M_2)
\end{aligned}$$

2) Vrai car  $CubeEmergent(r_2, r_1)$  est un espace convexe ; par conséquent tout tuple  $t \in S$  spécialise au moins un tuple  $v \in G$ .  $\square$

**Remarque 5** - Cette nouvelle caractérisation utilisant les minimaux transversaux cubiques s'applique, dans un contexte binaire, aussi bien pour le calcul des motifs émergents (Dong *et al.*, 2005), que pour celui des bordures des motifs contraints selon une conjonction hybride (Raedt *et al.*, 2001) en utilisant le concept classique de transversal (Eiter *et al.*, 1995, Berge, 1989).

$M_1$	(Roman, Marseille, Gallimard) (Scolaire, Paris, Hachette)
-------	--

**Tableau 7.** Ensemble  $M_1$  des tuples maximaux fréquents de  $DOCUMENT_1$

$M_2$	(Scolaire, Marseille, Gallimard) (Scolaire, Paris, Hachette) (Roman, Marseille, Gallimard) (Essai, Paris, ALL)
-------	---

**Tableau 8.** Ensemble  $M_2$  des tuples maximaux fréquents de  $DOCUMENT_2$

*Exemple 15* - Considérons les relations  $r_1 = DOCUMENT_1$  et  $r_2 = DOCUMENT_2$ . Les ensembles  $M_1$  et  $M_2$  sont donnés dans les tableaux 7 et 8.

Les bordures de  $CubeEmergent(r_2, r_1)$  avec  $MinSeuil_1 = 1/3$  et  $MinSeuil_2 = 1/5$  sont présentées dans les tableaux 9 et 10. Grâce à ces deux bordures, nous pouvons affirmer que les livres scolaires se sont mieux vendus à Marseille dans la deuxième relation car (Scolaire, Marseille, ALL) généralise le tuple (Scolaire, Marseille, Hachette) appartenant à  $S$  et il est généralisé par (Scolaire, Marseille, ALL) appartenant à  $G$ . En revanche, on ne peut rien affirmer pour le tuple (Scolaire, ALL, Gallimard) car il ne spécialise aucun tuple de  $G$ .

$S$	(Scolaire, Marseille, Gallimard) (Roman, Marseille, Gallimard) (Essai, Paris, ALL)
-----	--

**Tableau 9.** Majorant  $S$  de  $CubeEmergent(DOCUMENT_2, DOCUMENT_1)$

$G$	(ALL, ALL, Gallimard) (Scolaire, Marseille, ALL) (Essai, ALL, ALL)
-----	--

**Tableau 10.** *Minorant  $G$  de CubeEmergent(DOCUMENT<sub>2</sub>, DOCUMENT<sub>1</sub>)*

## 6. Conclusion

Nous avons, dans cet article, passé en revue différentes déclinaisons du concept de cube de données en leur ajoutant une nouvelle variation : les cubes émergents. En mettant en évidence les renversements de tendances ou, plus précisément, leurs évolutions dans des proportions significatives pour le décideur, les cubes émergents apportent de nouvelles connaissances particulièrement pertinentes pour comparer deux jeux de données. Leur représentation compacte et leur calcul efficace en font des candidats de choix pour toutes les applications d'analyse multidimensionnelle de flots de données. En effet, les utilisateurs de telles applications dynamiques cherchent expressément à connaître toute évolution de tendances pour être à même d'y réagir en temps réel.

Outre la proposition de ce nouveau type de cube, nous avons défini une structure unificatrice, le cube convexe, qui est un cadre formel et générique permettant de caractériser, de manière simple et solide, différentes variantes de cubes de données, trop souvent perçus comme les résultats de requêtes ou d'algorithmes et non comme des concepts. Nous nous sommes attachés à mettre en regard ces deux perceptions. Il résulte de ce travail une caractérisation homogène des divers types de cubes examinés, une classification qui se veut didactique pour que l'utilisateur choisisse la variante de cube adaptée à ses besoins mais surtout la possibilité d'une représentation compacte, solidement établie pour la structure générique du cube convexe et que nous montrons applicable à ses déclinaisons spécifiques que sont le cube iceberg, le cube intervalaire, le cube différentiel, le cube émergent et le cube de données lui-même. Quelle que soit la variante de cube considérée, la représentation par bordure obtenue est, à l'heure actuelle, la meilleure, *i.e.* la plus petite possible et donc, dans des contextes où ces considérations sont cruciales, la moins coûteuse à calculer (d'autant qu'il existe des algorithmes ayant prouvé leur efficacité) et la moins coûteuse à matérialiser.

## 7. Bibliographie

- Bayardo R., « Efficiently Mining Long Patterns from Databases », *Proceedings of the International Conference on Management of Data, SIGMOD*, p. 85-93, 1998.
- Berge C., *Hypergraphs : combinatorics of finite sets*, North-Holland, Amsterdam, 1989.
- Beyer K., Ramakrishnan R., « Bottom-Up Computation of Sparse and Iceberg CUBEs », *Proceedings of the International Conference on Management of Data, SIGMOD*, p. 359-370, 1999.

- Casali A., « Mining Borders of the Difference of Two Datacubes », *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery, DaWaK*, p. 391-400, 2004.
- Casali A., Cicchetti R., Lakhal L., « Cube Lattices : a Framework for Multidimensional Data Mining », *Proceedings of the 3rd SIAM International Conference on Data Mining, SDM*, p. 304-308, 2003a.
- Casali A., Cicchetti R., Lakhal L., « Extracting Semantics from Datacubes using Cube Transversals and Closures », *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, p. 69-78, 2003b.
- Casali A., Cicchetti R., Lakhal L., « Treillis cubes contraintes pour la fouille de bases de données multidimensionnelles », *Technique et Science Informatiques, TSI*, vol. 22 (10/2003), p. 1325-1352, 2003c.
- Dong G., Li J., « Mining border descriptions of emerging patterns from dataset pairs », *Knowledge Information System*, vol. 8 (2), p. 178-202, 2005.
- Eiter T., Gottlob G., « Identifying The Minimal Transversals of a Hypergraph and Related Problems », *SIAM Journal on Computing*, vol. 24(6), p. 1278-1304, 1995.
- Ganter B., Wille R., *Formal Concept Analysis : Mathematical Foundations*, Springer, 1999.
- Gouda K., Zaki M., « Efficiently Mining Maximal Frequent Itemsets », *Proceedings of the 1st IEEE International Conference on Data Mining, ICDM*, p. 3163-170, 2001.
- Gray J., Chaudhuri S., Bosworth A., Layman A., Reichart D., Venkatrao M., Pellow F., Pirahesh H., « Data Cube : A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals », *Data Mining and Knowledge Discovery*, vol. 1(1), p. 29-53, 1997.
- Gunopulos D., Mannila H., Khardon R., Toivonen H., « Data mining, hypergraph transversals, and machine learning », *Proceedings of the 16th Symposium on Principles of Database Systems, PODS*, p. 209-216, 1997.
- Han J., Chen Y., Dong G., Pei J., Wah B. W., Wang J., Cai Y. D., « Stream Cube : An Architecture for Multi-Dimensional Analysis of Data Streams », *Distributed and Parallel Databases*, vol. 18(2), p. 173-197, 2005.
- Han J., Kamber M., *Data Mining : Concepts and Techniques*, Morgan Kaufmann, 2001.
- Lakshmanan L., Pei J., Han J., « Quotient Cube : How to Summarize the Semantics of a Data Cube », *Proceedings of the 28th International Conference on Very Large Databases, VLDB*, p. 778-789, 2002.
- Mitchell T. M., « Generalization as Search », *Artificial Intelligence*, vol. 18(2), p. 203-226, 1982.
- Pei J., Han J., « Constrained Frequent pattern Mining : A Pattern-Growth View », *SIGKDD Explorations*, vol. 4(1), p. 31-39, 2002.
- Raedt L., Kramer S., « The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding », *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI*, p. 853-862, 2001.
- Vel M., *Theory of Convex Structures*, North-Holland, Amsterdam, 1993.